Web Tools for Rapid Experimental Visualization Prototyping

Jonathan W. Decker and Mark A. Livingston

Naval Research Laboratory, 4555 Overlook Ave SW, Washington, DC, USA

ABSTRACT

Quite often a researcher finds themselves looking at spreadsheets of high-dimensional data generated by experimental models and user studies. We can use analysis to challenge or confirm hypothesis, but unexpected results can easily be lost in the shuffle. For this reason, it would be useful to visualize the results so we can explore our data and make new discoveries.

Web browsers have become increasingly capable for creating complex, multi-view applications. Javascript is quickly becoming a de facto standard for scripting, online and offline. This work demonstrates the use of web technologies as a powerful tool for rapid visualization prototyping.

We have developed two prototypes: One for high-dimensional results of the abELICIT - multi-agent version of the *ELICIT* platform tasked with collaborating to identify the parameters of a pending attack. Another prototype displays responses to a user study on the effectiveness of multi-layer visualization techniques. We created coordinated multiple views prototypes in the Google Chrome web browser written in Javascript, CSS and HTML. We will discuss the benefits and shortcomings of this approach.

Keywords: web tools, experimental visualization

1. INTRODUCTION

Experimental models and user studies often produce tables of results that are just large enough to be understood in a spreedsheet. We obviously use analysis such as ANOVA to identify statistically significant differences from structured studies, but there's something to be said for being able to visually skim and dig down into individual records or even summaries in cases where ANOVA fails to produce desired insights. However, since experiment results are often high-dimensional and specialized to the task, they do not often map to well-known visual lexicons. So in order to create a focus-plus-context visualization of experiment results, we often need to build one from scratch, tailored to the specific results in question.

We propose that the evolving world of web technologies is suitable for prototype development of visualizations of experimental results. Modern web browsers have a multitude of built-in methods for drawing interactive elements, such as SVG and HTML 5 canvas. Also, developers can leverage powerful, existing APIs for visualizations, such as the Google Visualization API and the Data-Driven Documents Javascript Library (d3js).[1] However, web browsers are bleeding-edge development environments, and APIs and languages are subject to change. Further, web prototypes are unlikely to support views for large data sets with hundreds of thousands of records. Nevertheless, experiment results tend to produce small to medium-sized data sets, and tools to aid in their analysis don't require lengthy shelf lives. Thus, it would seem that this technology is well-suited for this type of flash-in-the-pan exploratory visualization.

This poster will discuss two prototypes: one for visualizing results of various parameterizations of models in the ELICIT framework, and another using results from a small-scale user study. For the ELICIT results visualization discussed in Section 3, we will give an overview of the problem space and explain the visualization we developed. For the user study prototype discussed in Section 4, we will describe the user study and the visualization of its results, and then we will make some observations about those results using the prototype. Finally, we will draw some conclusions about our experiences with using this approach to visualization design and discuss future work.

Report Documentation Page	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE **FEB 2013**	2. REPORT TYPE	3. DATES COVERED **00-00-2013 to 00-00-2013**

4. TITLE AND SUBTITLE **Web Tools for Rapid Experimental Visualization Prototyping**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Naval Research Laboratory,4555 Overlook Ave., SW,Washington,DC,20375**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR’S ACRONYM(S)
	11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

SPIE Visualization and Data Analysis, vol. 8654, Burlingame, CA, 4-6 Feb 2013.

14. ABSTRACT

Quite often a researcher finds themselves looking at spreadsheets of high-dimensional data generated by experi- mental models and user studies. We can use analysis to challenge or confirm hypothesis, but unexpected results can easily be lost in the shuffle. For this reason, it would be useful to visualize the results so we can explore our data and make new discoveries. Web browsers have become increasingly capable for creating complex, multi-view applications. Javascript is quickly becoming a de facto standard for scripting, online and offline. This work demonstrates the use of web technologies as a powerful tool for rapid visualization prototyping. We have developed two prototypes: One for high-dimensional results of the abELICIT - multi-agent version of the ELICIT platform tasked with collaborating to identify the parameters of a pending attack. Another prototype displays responses to a user study on the effectiveness of multi-layer visualization techniques. We created coordinated multiple views prototypes in the Google Chrome web browser written in Javascript, CSS and HTML. We will discuss the benefits and shortcomings of this approach.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**7**	

2. RELATED WORK

The primary view in our prototypes is a nested graph of unlabeled colored grids which correspond to individual trials of the data set, and could be considered a hierarchical form of a spatial technique called *Attribute Blocks*.[2] Attribute blocks consist of a cluster of shapes or a divided shape to represent multiple values at sample points; we chose squares. Each attribute may be visualized with a continuous variable, such as color or intensity; we use heat maps that differ between our prototypes. Variables are separated by their location within the cluster or shape. Dynamically changing the configuration of the elements and the size and origin of the individual components enables the user to interactively control which variables are presented.

Parallel Coordinates is a common approach for visualizing numerous dimensions of numerical data.[3] Each basis dimension gets an axis in the set of axes; these are drawn parallel to each other. Typical implementations of parallel coordinates use vertical axes for the domain. A single point is then drawn as a polyline connecting the basis axes; this polyline intersects each axis at the value for that dimension in the point. For the purposes of this prototype, we have added a series of black boxes below the axes to represent unset data values in each category. Other important issues in parallel coordinates include the ordering of the axes (since correlations are most easily seen between neighboring axes) and the selection of the basis dimensions to use. Our prototype simply displays all of the available dimensions in the order they are presented in the data set.

One of the prototypes in this work concerns the results of user study we conducted on existing visualization techniques. Figure 9 shows a image composite of six independent layers using Data-Driven Spots.[4] This is one of several different techniques our study covered for layering independent layers and timesteps. Since these techniques are not relevant to the work in this paper, we will avoid summarizing them all for the sake of clarity and focus. Please see our forthcoming paper[5] for details.

3. ELICIT PROTOTYPE

For our first prototype, we were tasked with visualizing results of a multi-agent simulation run with various parameters. The followings sections summarize the problem area and visualization we designed to communicate the results.

3.1 Problem Description

For this work, we developed a prototype layout in order to convey a unique data set. This data comes from experimentation on the *ELICIT* platform. ELICIT stands for Experimental Laboratory for Investigating Collaboration, Information-sharing, and Trust. This platform provides an environment where humans or computer agents are tasked with discovering the precise parameters of an upcoming terrorist attack. These entities have accomplished their mission when they agree on correct information about the attack within a given length of time. When the platform is completely agent driven this is referred to as *abELICIT* [6, Chapter 19]. The data set we were tasked with visualizing is the output of a series of trials where abELICIT was run using several values for a number of configuration parameters. Each data entry includes the value for each parameter and a series of performance metrics of each agent in the system for that trial.

3.2 Visualization Design

Visualizing this data is difficult because each trial produces eight performance metrics for four agent roles and an aggregate across all roles. Additionally, each of these trials has the context of its input parameters. We decided to preserve this information by representing each trial as an unlabeled grid of colored rectangles. Each rectangle represents the value of a single metric for a single role in the group. Figure 1 shows an interface that is displayed above the visualization and conveys how the entries are organized within each trial table. In the top right of the same figure is a gradient that shows the colored mapping for each entry in the table. On the right of the figure is an example of what a trial looks like represented thusly. When the entry is black, this means the value was unset in the data. The example trial table is filtered so that the Avg. Task Performance row and the Team Leader agent results are not shown. The user can click the names in this view to filter the trial results interactively.

Each trial table is nested within additional tables to show its position within the parameter space. Figure 2 shows a single cell of the largest table. The labels on the outside correspond to the value of specific parameters

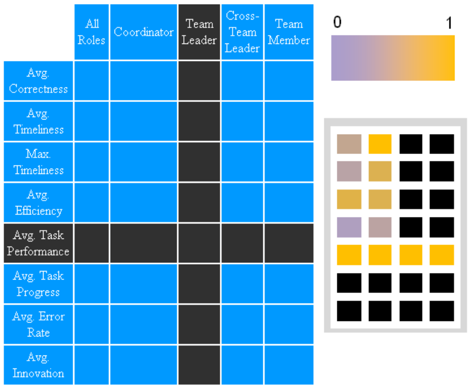


Figure 1. On the left, the interface for filtering the rows and columns of individual trial cells. On the right, a corresponding filtered trial cell, with the Team Leader column and the Avg. Task Performance row removed.

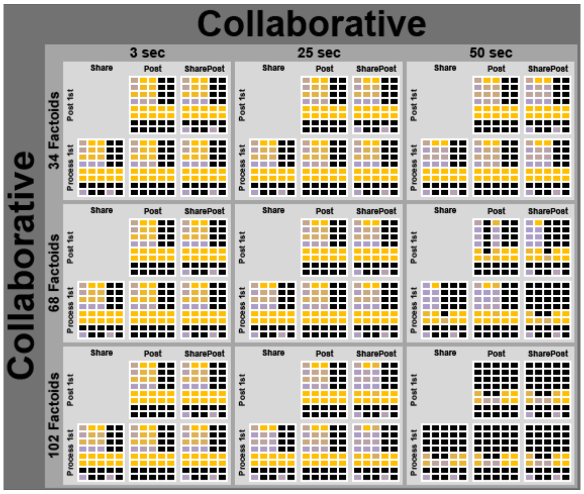
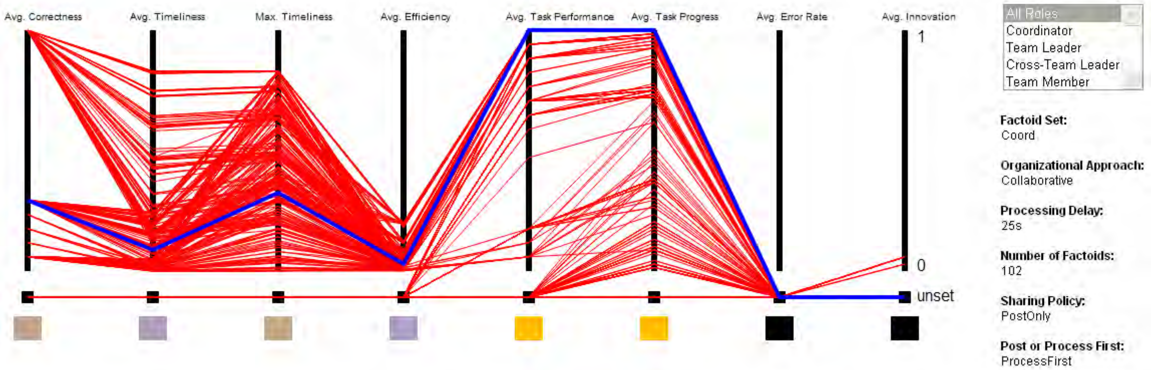


Figure 2. Top right corner of the nested grid.



Factoid Set:
Coord

Organizational Approach:
Collaborative

Processing Delay:
25s

Number of Factoids:
102

Sharing Policy:
PostOnly

Post or Process First:
ProcessFirst

Figure 3. Parallel coordinates view in web prototype, which shows the value of each attribute for one role at a time, comparing a selected trial (blue line) with all other trials produced by the model.

for the nested trials. The missing trial cell within the third table represents an impossible setting for two parameters. The tables were organized so that the overall visualization was mostly square, but otherwise the mapping is arbitrary. Table 1 is Factoid Set by Organization Approach, Table 2 is Processing Delay by Number of Factoids and Table 3 is Sharing Policy by Post or Process First. As previously mentioned, we will not be able to go over the specifics of how these parameters relate to the framework.

We began our investigation by examining the parallel coordinates and derivative methods. One major design decision was what to use as the domain axes in the parallel coordinates plot. We made an initial visualization that used the input domains (i.e. the configuration parameters) as the axes. The parameter values are either a discrete set of numerical values or categorical values (different strategies, etc.). Thus this initial visualization (Figure 4) was a Parallel Sets[7] extension of parallel coordinates, created using an existing tool called ParSets (http://code.google.com/p/parsets/). We found that these sets broke down into nearly all the possible combinations, which resulted in an incoherent visual representation of the data. Thus we began to investigate using the dependent variables of the simulation (i.e. the eight performance metrics defined for the ELICIT platform) for the domain axes in a parallel coordinates plot.

The resulting parallel coordinates plot is displayed beside the trial layout grid and can be seen in Figure 3. This view shows the results for one role at a time for all trials, and the user can select a trial cell in the nested tables to see it appear as a thick, blue polyline on this chart. Under each axis is a box for unset values, which are common in the data set (Notice in Figure 3 that Avg. Error Rate is never set). Below those boxes is a color that corresponds to the colors in the trial cell for the selected trial. The final interaction not shown here is that

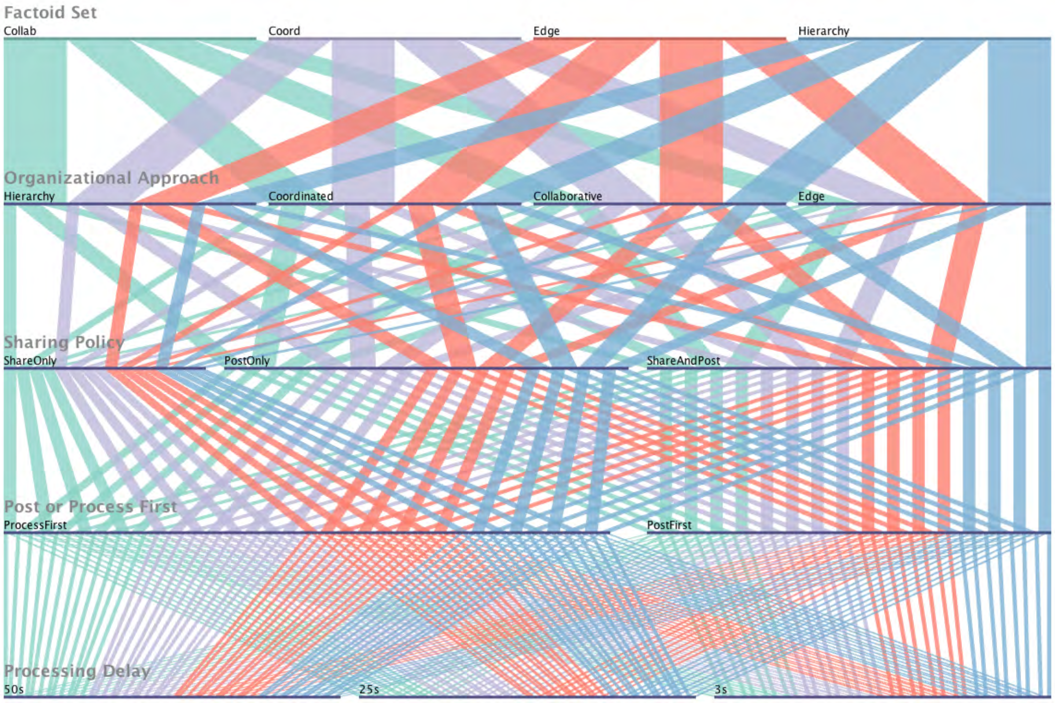


Figure 4. Parallel sets view of the first five parameters of the ELICIT data set, created with the existing ParSets tool. From this view we could see that these parameters break down into nearly all combinations.

we can mouse-over any entry in the nested tables and and see what role and metric it references, as well as the values of its attributes.

The initial Parallel Sets view also helped us realize we would need a different way to represent the parameter context, since this view leaves little screen space for the individual model results. This lead us to our table representation of the hierarchical parameter space.

3.3 Implementation

This work demonstrates that web technologies are viable for offline development of visualization. We do not propose any new framework or design patterns that do not already exist in this space. We implemented these prototypes in HTML, CSS, and Javascript. The parallel coordinates view is drawn with HTML5 Canvas, and the grids are created using HTML table tags. Javascript handles constructing the views from the data, and rebuilding them on user interaction. CSS is used to in concert with the HTML to define the page elements. The Javascript is organized into a series of functions that handle aggregating the data, building the tables, and rewriting the page on user input. The layout was built for workstations with large 2560x1600 monitors, so the visualization does not conform to the size of the browser window. Using responsible design would only become necessary if the audience of the visualization grow beyond our workgroup.

4. USER STUDY PROTOTYPE

For our second prototype, we used results obtained from a user study studying multivariate visualization. The following sections will briefly describe this study and the visualization we designed to communicate the results.

4.1 Problem Description

We have been engaged in a series of iterative usability studies of existing multivariate visualization techniques. The visualization we will discuss here encodes results from our most recent study. For the sake of clarity, we will only discuss details relative to the results demonstrated by the experimental visualization. Please refer to our paper detailing the study for more information.[5]

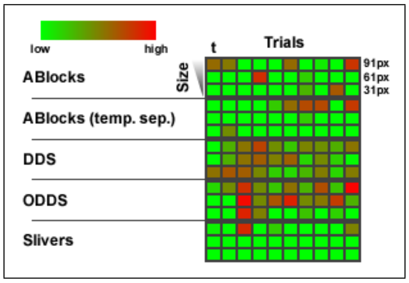


Figure 5. Legend describing user grid.

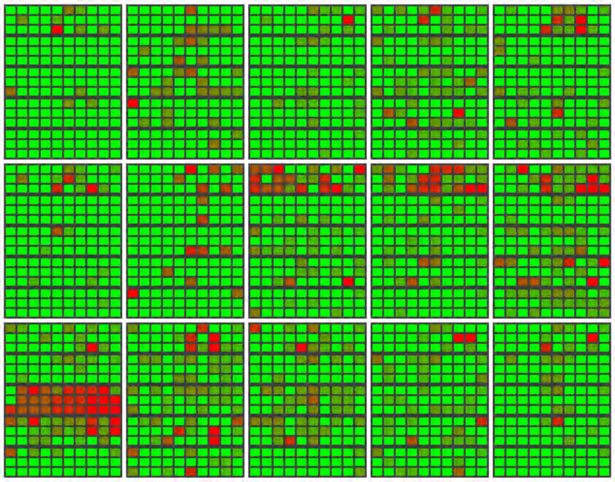


Figure 6. Grid of user results for error per trial.

The study looked at five separate techniques, each of which reduces a series of time-varying, multi-layer spatial data into a reduced set of image composites. The data set concerns spatial fields of white, identically-sized squares randomly placed on a black background. Each series has six layers of fields per timestep, with six discrete timesteps. User were asked to find the timestep and position in 2D space where the most squares from different layers overlap. Squares within the same timestep do not overlap. The techniques we examined composite the layers at each timestep into single images. There are also two techniques which composite all the timesteps together. The end result is that instead of looking at 36 images, users only need to look at 1-6 images to complete their task, depending on the technique being evaluated.

The users are shown 30 examples of the same technique, 10 examples per target size. For each question, or example of the data set, we measure the time the user took, the distance the user was from the target square, and an error value. This error is related to actual number of overlapping fields at the position and timestep the user selected.

With the results, we conduct ANOVA to determine statistically significant differences between how well subjects performed with each technique. For this poster, we explore the results in a exhaustive visual layout to see what new insights we can discover.

4.2 Visualization Design

Much like the ELICIT Prototype described above, we use a nested table of unlabeled cells to display all of the results at once. However, instead of have each innermost cell display a series of different results, we create a series of separate grids for each result. This makes it easier to compare like values. The legend for the grid can be seen in Figure 5. This legend reveals the decomposition of the glyph for one user in the study. The glyph is divided into five rows of smaller grids, one for each technique we tested. Each of these grids is further divided into 3 rows and 10 columns of colored squares. The rows indicate the size of the targets in the trial (largest to smallest: 91px, 61px, 31px) and the columns indicate repeated trials with the same parameters. The first column in every grid is the tutorial question. We will refer to these inner grids as the *technique grids*.

Using this method, we created three grids based on the user results. The grid in Figure 6 shows an error metric used in the evaluation. The grid in Figure 7 shows the time the user took relative to the largest time interval among their responses. The grid in Figure 8 shows the distance the user's selection was from the target square.

Also like our first prototype, we added an interactive element. Each of the trials in our user study involve a series of images, and the results in the grids relate to points within these images. An analyst with this prototype visualization can click any of the grid squares and see the corresponding image(s) presented to the user for that trial. We also overlay a cursor for each user response in the study for that image. Figure 9 shows timestep 4

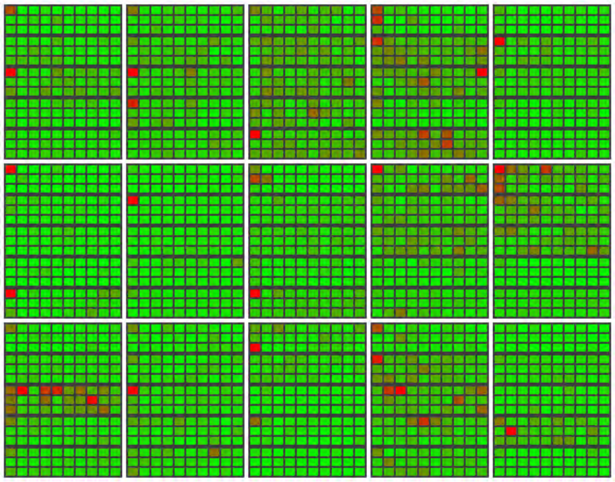


Figure 7. Grid of user results for time taken per trial.

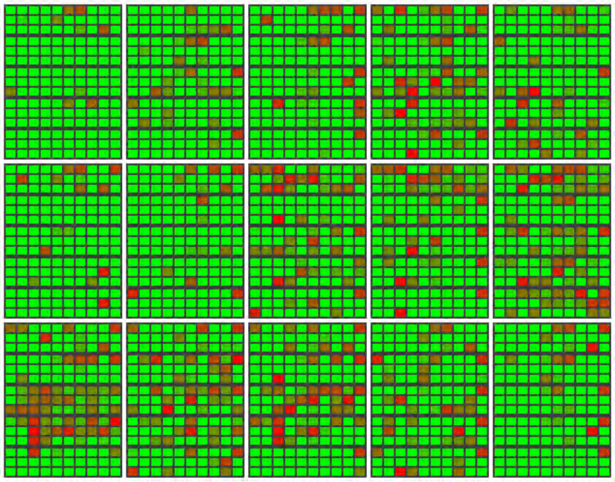


Figure 8. Grid of user results for distance from target.

and timestep 6 of question 35 rendered using the DDS technique. The blue cursor places the center of the target square, which is in timestep 4 in this example. The prototype can switch between timestep images by using the arrow keys. The red cursor shows the user selection for the user currently focused on by the prototype. We'll discuss observations about what can be seen in this view in the next section.

4.3 Observations

Users generally took more time on the first tutorial question they were given, which was always used the 91-pixel target. This can be seen in Figure 7, indicated by a red square in the top left corner of each technique grid. Notice the user represented by the grid in the end of the middle row. We can see that they spent more time on the tutorial questions (first row) for attribute blocks than they did for the following questions.

The ANOVA computations generate summary tables, from which we found that both Oriented Slivers and the temporally separated Attribute Blocks technique performed very well. This can be seen in the error and distance grids (Figure 6 and Figure 8, respectively) by the lack of red color among the squares in second and fifth inner grid within each user grid. This confirms that we can see results from the ANOVA in this visualization; however, since these results are easily obtained through the ANOVA, we focus on results found through this visualization.

We also made several new observations from viewing the data in this manner. Look at Figure 6 and Figure 7 and take note of the the user represented by the bottom-left grid. Notice the large values in the DDS technique grid for both time and error. This user clearly did very poorly on this technique. Using the focus view mentioned above, we observe the user is seldom clicking on anything other than the black background; their answers seem random. The order users see techniques is determined by a Latin square, and this user received the DDS technique first. It is clear from exploring the data through this visualization that this user was confused by the interface, and was not sure at first what they were being asked to do. We can also discover they they did fine on the first two tutorial questions, but unlike many other users, their time did not improve over the course of the tutorial. This is some indication that they did not understand the task. The DDS technique did not do as well as we expected overall; this visualization prompted us to measure the effect this user's poor performance had on the assessment of DDS in the study. When conducting a new ANOVA with this subject's data removed, we found a 31% drop in the mean error computed for DDS, implying that this user's performance severely (and adversely) impacted our assessment of the technique.

We also observed that Question 35 was hard across all techniques. This can be seen clearly in Figure 9, which shows the DDS renderings of two of the six timesteps of the question. Although the target is on the timestep represented by the left image the majority of users clicked within a mass of distractors in the right image. Notice the top-right square of every technique grid in the error grid. Many users were continuously challenged by this configuration, regardless of how the data was presented.

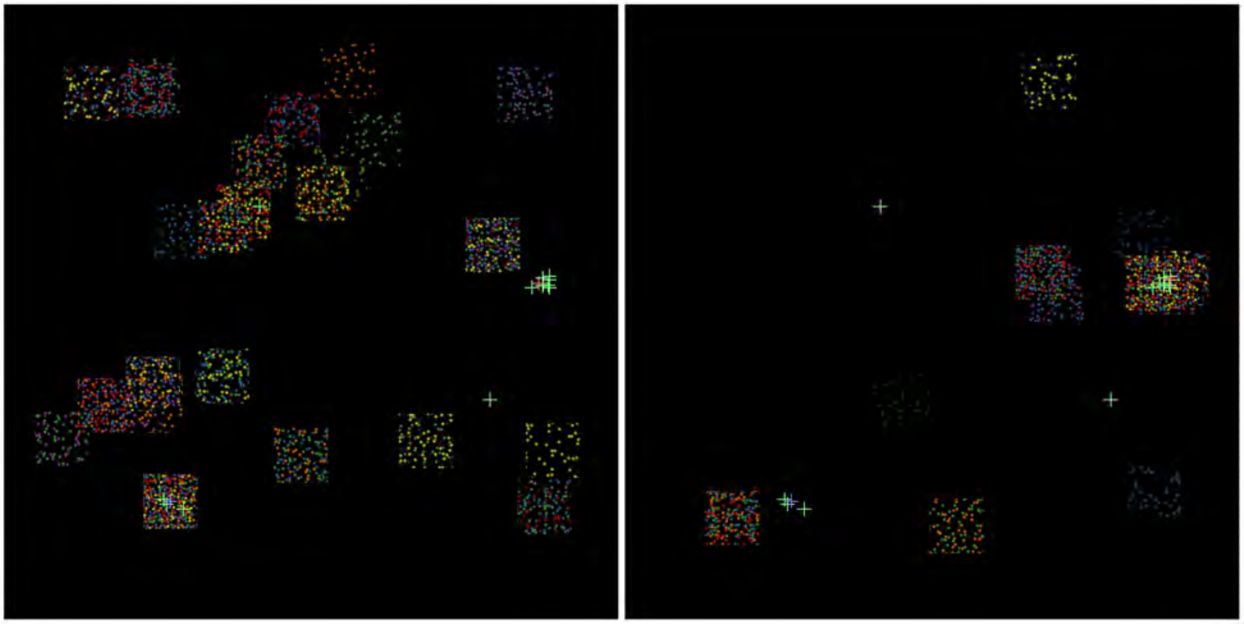


Figure 9. Two timesteps from our multi-layer study results viewer. The crosses indicate the places where users clicked and the location of the target. Here we can see that most users selected a dense region of several overlapped distractors in the right-hand timestep while the actual solution was near the lower-left in the left-hand timestep.

5. CONCLUSION AND FUTURE WORK

We have demonstrated the implementation of two prototypes within a web browser using Javascript, HTML and CSS. We found it exceptionally easy to implement our prototypes in this environment, and we recommend this approach for experimental visualization. Web browsers provide an environment for rapid development of rectilinear layouts, which is useful for a large class of visualizations. There is also a multitude of useful APIs that can be leveraged, and browsers are constantly evolving and adding features. For these reasons, we see browsers as a strong avenue for prototype development moving forward, at least for multi-view layouts of medium-sized datasets.

In future work, we would like to take a larger look at libraries available to the the web programmer for designing visualizations, such as the Data-Driven Documents Javascript Library (d3js).[1] A major weakness of our prototypes is that we don't given the user any control over the range used to normalize the data values. This makes it hard to observe much more than outliers within the data. We'd like to add a range window in a future iteration so that we can remove outliers data points and see more subtle variance.

REFERENCES

1. M. Bostock, "Data-Driven Documents (d3js.org)." http://d3js.org/, August 2012.
2. J. Miller, "Attribute blocks: Visualizing multiple continuously defined attributes," *Computer Graphics and Applications, IEEE* **27**, pp. 57 –69, may-june 2007.
3. M. d'Ocagne, *Coordonnées Parallèles et Axiales: Méthode de transformation géométrique et procédé nouveau de calcul graphique déduits de la considération des coordonnées parallèlles*, Gauthier-Villars, 1885.
4. A. A. Bokinsky, *Multivariate Data Visualization with Data-driven Spots.* PhD thesis, The University of North Carolina at Chapel Hill, 2003.
5. M. A. Livingston, J. W. Decker, and Z. Ai, "Evaluation of multivariate visualizations for time-varying data," in *In SPIE Visualization and Data Analysis*, 2013.
6. D. S. Alberts, *The Agility Advantage: A Survival Guide*, ch. 19, pp. 287–304. CCRP Publication Series, 2011.
7. R. Kosara, F. Bendix, and H. Hauser, "Parallel sets: Interactive exploration and visual analysis of categorical data," *IEEE Transactions on Visualization and Computer Graphics* **12**, pp. 558–568, July 2006.